Statistical Applications in Genetics and Molecular Biology

Volume 6, *Issue* 1 2007 *Article* 25

Super Learner

Mark J. van der Laan* Eric C. Polley†

Alan E. Hubbard‡

*University of California, Berkeley, laan@stat.berkeley.edu

†University of California, Berkeley, ecpolley@berkeley.edu

‡University of California, Berkeley, hubbard@stat.berkeley.edu

Super Learner

Mark J. van der Laan, Eric C. Polley, and Alan E. Hubbard

Abstract

When trying to learn a model for the prediction of an outcome given a set of covariates, a statistician has many estimation procedures in their toolbox. A few examples of these candidate learners are: least squares, least angle regression, random forests, and spline regression. Previous articles (van der Laan and Dudoit (2003); van der Laan et al. (2006); Sinisi et al. (2007)) theoretically validated the use of cross validation to select an optimal learner among many candidate learners. Motivated by this use of cross validation, we propose a new prediction method for creating a weighted combination of many candidate learners to build the super learner. This article proposes a fast algorithm for constructing a super learner in prediction which uses V-fold cross-validation to select weights to combine an initial set of candidate learners. In addition, this paper contains a practical demonstration of the adaptivity of this so called super learner to various true data generating distributions. This approach for construction of a super learner generalizes to any parameter which can be defined as a minimizer of a loss function.

KEYWORDS: cross-validation, loss-based estimation, machine learning, prediction

1 Introduction

Numerous methods exist to learn from data the best predictor of a given outcome based on a sample of n independent and identically distributed observations $O_i = (Y_i, X_i)$, Y_i the outcome of interest, and X_i a vector of input variables, $i = 1, \ldots, n$. A few examples include decision trees, neural networks, support vector regression, least angle regression, logic regression, poly-class, Multivariate Adaptive Regression Splines (MARS), and the Deletion/Substitution/Addition (D/S/A) algorithm. Such learners can be characterized by the mechanism used to search the parameter space of possible regression functions. For example, the D/S/A algorithm (Sinisi and van der Laan, 2004) uses polynomial basis functions, while logic regression (Ruczinski et al., 2003) constructs Boolean expressions of binary covariates. The performance of a particular learner depends on how effective its searching strategy is in approximating the optimal predictor defined by the true data generating distribution. Thus, the relative performance of various learners will depend on the true data-generating distribution. In practice, it is generally impossible to know *a priori* which learner will perform best for a given prediction problem and data set. To solve the problem, some researchers have proposed combining learners in various methods and have exhibited better performance over a single candidate learner (Freund et al., 1997; Hansen, 1998), but there is concern that these methods may over-fit the data and may not be the optimal way to combine the candidate learners.

The framework for unified loss-based estimation (van der Laan and Dudoit, 2003) suggests a solution to this problem in the form of a new learner, termed the "super learner". In the context of prediction, this learner is itself a prediction algorithm, which applies a set of candidate learners to the observed data, and chooses the optimal learner for a given prediction problem based on cross-validated risk. Theoretical results show that such a super learner will perform asymptotically as well as or better than any of the candidate learners (van der Laan and Dudoit, 2003; van der Laan et al., 2006).

To be specific, consider some candidate learners. *Least Angle Regression* (LARS) (Efron et al., 2004) is a model selection algorithm related to the lasso. *Logic Regression* (Ruczinski et al., 2003) is an adaptive regression methodology that attempts to construct predictors as Boolean combinations of binary covariates. The D/S/A algorithm (Sinisi and van der Laan, 2004) for polynomial regression data-adaptively generates candidate predictors as polynomial combinations of continuous and/or binary covariates, and is avail-

Method	R Package	Authors
Least Angle Regression	lars	Hastie and Efron
Logic Regression	LogicReg	Kooperberg and Ruczinski
D/S/A	DSA	Neugebauer and Bullard
Regression Trees	rpart	Therneau and Atkinson
Ridge Regression	MASS	Venables and Ripley
Random Forests	randomForest	Liaw and Wiener
Adaptive Regression Splines	polspline	Kooperberg

Table 1: R Packages for Candidate Learners. R is available at http://www.r-project.org

able as an `R` package at http://www.stat.berkeley.edu/users/laan/Software/. *Classification and Regression Trees* (CART) (Breiman et al., 1984) builds a recursive partition of the covariates. Another candidate learner is random forests Breiman (2001), which is a random bootstrap version of the regression tree. *Ridge Regression* (Hoerl and Kennard, 1970) minimizes a penalized least squares with a penalty on the $L2$ norm of the parameter vector. *Multivariate Adaptive Regression Splines* (MARS) Friedman (1991) is an automated model selection algorithm which creates a regression spline function. Table 1 contains citations of R packages for each of the candidate learners. All of these methods have the option to carry out selection using v-fold cross-validation. The selected fine-tuning parameter(s) can include the ratio of the $L1$ norm of the coefficient vector in LARS to the norm of the coefficient vector from least squares; the number of logic trees and leaves in Logic Regression; and the number of terms and a complexity measure on each of the terms in D/S/A.

Cross-validation divides the available *learning* set into a *training* set and a *validation* set. Observations in the training set are used to construct (or *train*) the learners, and observations in the validation set are used to assess the performance of (or *validate*) these learners. The cross-validation selector selects the learner with the best performance on the validation sets. In v-fold cross-validation, the learning set is divided into v mutually exclusive and exhaustive sets of as nearly equal size as possible. Each set and its complement play the role of the validation and training sample, respectively, giving v splits of the learning sample into a training and corresponding validation sample. For each of the v splits, the estimator is applied to the training

set, and its risk is estimated with the corresponding validation set. For each learner the v risks over the v validation sets are averaged resulting in the so-called *cross-validated risk*. The learner with the minimal cross-validated risk is selected.

It is helpful to consider each learner as an algorithm applied to empirical distributions. Thus, if we index a particular learner with an index k, then this learner can be represented as a function $P_n \rightarrow \hat{\Psi}_k(P_n)$ from empirical probability distributions P_n to functions of the covariates. Consider a collection of $K(n)$ learners $\hat{\Psi}_k$, $k = 1, \ldots, K(n)$, in parameter space $\mathbf{\Psi}$. The super learner is a new learner defined as

$$\hat{\Psi}(P_n) \equiv \hat{\Psi}_{\hat{K}(P_n)}(P_n),$$

where $\hat{K}(P_n)$ denotes the cross-validation selector described above which simply selects the learner which performed best in terms of cross-validated risk. Specifically,

$$\hat{K}(P_n) \equiv \arg\min_k E_{B_n} \sum_{i, B_n(i)=1} (Y_i - \hat{\Psi}_k(P^0_{n,B_n})(X_i))^2,$$

where $B_n \in \{0,1\}^n$ denotes a random binary vector whose realizations define a split of the learning sample into a training sample $\{i : B_n(i) = 0\}$ and validation sample $\{i : B_n(i) = 1\}$. Here P^1_{n,B_n} and P^0_{n,B_n} are the empirical probability distributions of the validation and training sample, respectively.

The aggressive use of cross-validation is inspired by the theorem 3.1 in van der Laan et al. (2006). The theorem is provided in the appendix.

The "oracle" selector is defined in Theorem 2 in the appendix as the estimator, among the $K(n)$ learners considered, which minimizes risk under the true data-generating distribution. In other words, the oracle selector is the best possible estimator given the set of candidate learners considered; however, it depends on both the observed data and P_0, and thus is unknown.

This theorem shows us that the super learner performs as well (in terms of expected risk difference) as the oracle selector, up to a typically second order term. Thus, as long as the number of candidate learners considered ($K(n)$) is polynomial in sample size, the super learner is the optimal learner in the following sense:

- If, as is typical, none of the candidate learners (nor, as a result, the oracle selector) converge at a parametric rate, the super learner performs asymptotically as well (in the risk difference sense) as the oracle selector, which chooses the best of the candidate learners.

- If one of the candidate learners searches within a parametric model and that parametric model contains the truth, and thus achieves a parametric rate of convergence, then the super learner achieves the almost parametric rate of convergence $\log n/n$.

Organization: The current article builds and extends this super learning methodology. In section 2 we will describe our new proposal for super learning, also using an initial set of candidate learners and cross-validation as above, but now allowing for semi-parametric families of the candidate learners, and formulating the minimization of cross-validated risk as another regression problem for which one can select an appropriate regression methodology (e.g involving cross-validation or penalized regression). This is an important improvement relative to our previous super learning proposal by 1) extending the set of initial candidate learners into a large family of candidate learners one obtains by combining the initial candidate learners according to a parametric or semi-parametric model, thereby obtain a potentially much more flexible learner, and 2) by controlling over-fitting of the cross-validated risk through the use of data adaptive regression algorithms using cross-validation or penalization itself. Importantly, these gains come at no cost regarding computing time. In Section 3 we investigate the practical performance of this new super learning algorithm based on simulated as well as a number of real data sets.

2 The proposed super learning algorithm

Suppose one observes n i.i.d. observations $O_i = (X_i, Y_i) \sim P_0$, $i = 1, \ldots, n$, and the goal is to estimate the regression $\psi_0(X) = E_0(Y \mid X)$ of $Y \in \mathcal{Y}$ on $X \in \mathcal{X}$. The regression can be defined as the minimizer of the expectation of the squared error loss function:

$$\psi_0 = \arg\min_{\psi} E_0 L(O, \psi),$$

where $L(O, \psi) = (Y - \psi(X))^2$. The proposed super learner immediately applies to any parameters that can be defined as minimizers of a loss function $L(O, \psi)$ over a parameter space $\mathbf{\Psi}$, but the article focuses on the prediction problem using the squared error loss function.

Let $\hat{\Psi}_j$, $j = 1, \ldots, J$, be a collection of J candidate learners, which represent mappings from the empirical probability distribution P_n into the parameter space $\mathbf{\Psi}$ consisting of functions of X.

The proposed super learner uses V-fold cross-validation. Let $v \in \{1, \ldots, V\}$ index a sample split into a validation sample $V(v) \subset \{1, \ldots, n\}$ and training sample (the complement of $V(v)$) $T(v) \subset \{1, \ldots, n\}$, where $V(v) \cup T(v) = \{1, \ldots, n\}$. Here we note that the union, $\cup_{v=1}^{V} V(v) = \{1, \ldots, n\}$, of the validation samples equals the total sample, and the validations samples are disjoint: $V(v_1) \cap V(v_2) = \emptyset$ for $v_1 \neq v_2$. For each $v \in \{1, \ldots, V\}$, let, $\psi_{njv} \equiv \hat{\Psi}_j(P_{nT(v)})$ be the realization of the j^{th}-estimator $\hat{\Psi}_j$ when applied to the training sample $P_{nT(v)}$.

For an observation i, let $v(i)$ denote the validation sample it belongs to, $i = 1, \ldots, n$. We now construct a new data set of n observations as follows: (Y_i, Z_i), where $Z_i \equiv (\psi_{njv(i)}(X_i) : j = 1, \ldots, J)$ is the vector consisting of the J predicted values according to the J estimators trained on the training sample $P_{nT(v(i))}$, $i = 1, \ldots, n$. Let $\mathcal{Z}$ be the set of possible outcomes for Z.

Minimum cross-validated risk predictor: Another input of this super learning algorithm is yet another user-supplied prediction algorithm $\tilde{\Psi}$ that estimates the regression $E(Y \mid Z)$ of Y onto Z based on the data set (Y_i, Z_i), $i = 1, \ldots, n$. For notational convenience, we will denote $\{(Y_i, Z_i) : i = 1, \ldots, n\}$ with $P_{n,Y,Z}$, so that $\tilde{\Psi}$ is a mapping from $P_{n,Y,Z}$ to $\tilde{\Psi}(P_{n,Y,Z}) : \mathcal{Z} \to \mathcal{Y}$, where the latter is a function from $\mathcal{Z}$ to $\mathcal{Y}$. We will refer to this algorithm $\tilde{\Psi}$ as the minimum cross-validated risk predictor since it aims to minimize the cross-validated risk, $\tilde{\psi} \to \sum_{i=1}^{n}(Y_i - \tilde{\psi}(Z_i))^2$, over a set of candidate functions $\tilde{\psi}$ from $\mathcal{Z}$ into $\mathcal{Y}$, although, we allow penalization or cross-validation to avoid over-fitting of this cross-validated risk criteria.

This now defines a mapping $\hat{\Psi}^*$ from the original data $P_n \equiv \{Y_i, X_i) : i = 1, \ldots, n\}$ into the predictor

$$\tilde{\Psi}\left(\{Y_i, Z_i = (\hat{\Psi}_j(P_{nT(v_i)})(X_i) : j = 1, \ldots, J)) : i = 1, \ldots, n\}\right)$$

obtained by applying the cross-validated risk minimizer $\tilde{\Psi}$ to $P_{n,Y,Z} = \{(Y_i, Z_i) : i = 1, \ldots, n\}$. Denote $\psi_n^* = \hat{\Psi}^*(P_n)$ as the actual obtained predictor when one applies the learner $\hat{\Psi}^*$ to the original sample P_n. We note that $\psi_n^* \in \mathbf{\Psi}^* \equiv \{f : \mathcal{Z} \to \mathcal{Y}\}$ is a function of Z into the outcome set $\mathcal{Y}$ for Y.

The super learner for a value X based on the data (i.e., P_n) is now given by

$$\hat{\Psi}(P_n)(X) \equiv \hat{\Psi}^*(P_n)((\hat{\Psi}_j(P_n)(X), j = 1, \ldots, J). \quad (1)$$

In words, the super learner of Y for a value X is obtained by evaluating the predictor $\psi_n^* = \hat{\Psi}^*(P_n)$ at the J predicted values, $\hat{\Psi}_j(P_n)(X)$, at X of the J

candidate learners. Figure 1 contains a flow diagram for the steps involved in the super learner.

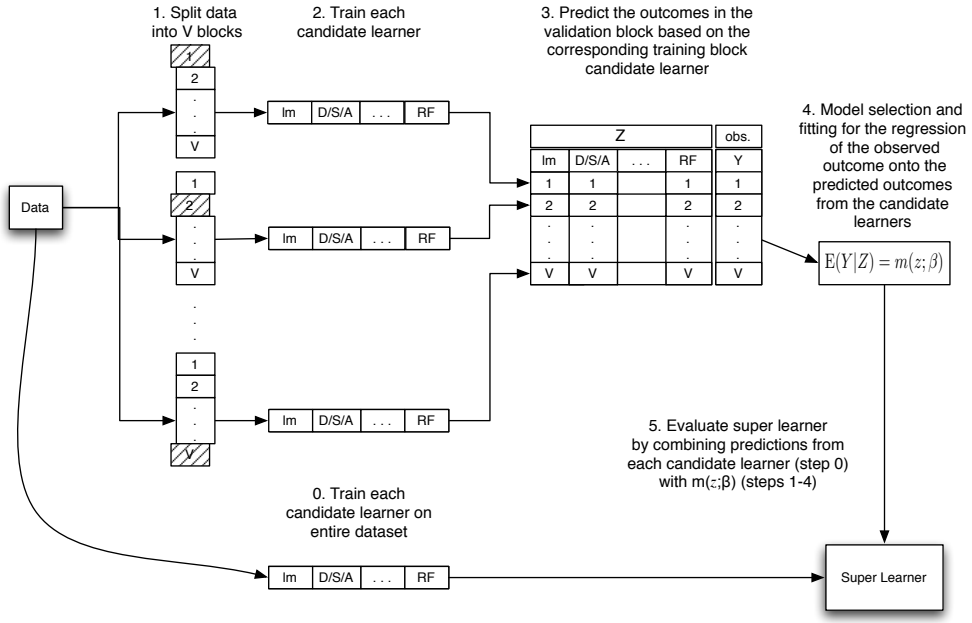

Figure 1: Flow Diagram for Super Learner

2.1 Specific choices of the minimum cross-validated risk predictor.

Parametric minimum cross-validated risk predictor: Consider a few concrete choices that aim to fit a regression of Y onto the J predicted values Z based on the corresponding training samples from (Y_i, Z_i), $i = 1, \ldots, n$ for the algorithm $\hat{\Psi}^*$. Define the cross-validated risk criteria:

$$R_{CV}(\beta) \equiv \sum_{i=1}^{n} (Y_i - m(Z_i \mid \beta))^2,$$

where one could use, for example, the linear regression model $m(z \mid \beta) = \beta z$. If $Y \in \{0, 1\}$, then one could use the logistic linear regression model

$m(z \mid \beta) = 1/(1 + \exp(-\beta z))$, if one allows predictions in the range of $[0, 1]$, or, if one wants a predictor mapping into $\{0, 1\}$, then we can choose $m(z \mid \alpha_0, \beta) \equiv I(1/(1 + \exp(-\beta z)) > \alpha_0)$ as the indicator that the logistic regression score exceeds a cut-off α_0. Let $\beta_n = \arg\min_\beta R_{CV}(\beta)$ be the least squares or MLE estimator, and let

$$\psi_n^*(z) \equiv m(z \mid \beta_n).$$

One could also estimate β with a constrained least squares regression estimator such as penalized L_1-regression (Lasso), penalized L_2 regression (shrinkage), where the constraints are selected with cross-validation, or one could restrict β to the set of positive weights summing up till 1.

Since the candidate learners are all trying to predict the same outcome Y there is a potential for collinearity or near collinearity in the predicted Z data set. In practice, the user could simply remove one of the troublesome candidate learners, which is often the default in most statistical regression software when collinearity is present. Near collinearity can make the interpretation of $\psi_n^*(z)$ difficult, especially if evaluating the magnitude of the parameters β_n. We recommend caution in interpretation of the parameter estimates but note that near collinearity should not effect the prediction accuracy of the super learner.

Data adaptive minimum cross-validated risk predictor: There is no need to restrict ψ_n^* to parametric regression fits. For example, one could define ψ_n^* in terms of the application of a particular data adaptive (machine learning) regression algorithm to the data set (Y_i, Z_i), $i = 1, \ldots, n$, such as CART, D/S/A, or MARS, among others. In fact, one could apply a super learning algorithm itself to estimate $E(Y \mid Z)$. In this manner one can let the data speak in order to build a good predictor of Y based on covariate vector Z based on (Y_i, Z_i), $i = 1, \ldots, n$.

Thus, this super learner is indexed, beyond the choice of initial candidate estimators, by a choice of minimum cross-validated risk predictor. As a consequence, the proposal provides a whole class of tools indexed by an arbitrary choice of regression algorithm (i.e., ψ_n^*) to map a set of candidate learners into a new cross-validated estimator (i.e. super learner). In particular, it provides a new way of using the cross-validated risk function, which goes beyond minimizing the cross-validated risk over a set of candidate learners.

3 Finite sample result and asymptotics for the super learner.

An immediate consequence of Theorem 2 above is the following result for the proposed super learner (1), which provides for the case that the minimum cross-validated risk predictor is based on a parametric regression model.

Theorem 1. *Assume $P((Y,X) \in \mathcal{Y} \times \mathcal{X}) = 1$, where $\mathcal{Y}$ is a bounded set in $\mathbb{R}$, and $\mathcal{X}$ is a bounded Euclidean set. Assume that the candidate estimators map into $\mathcal{Y}$: $P(\hat{\Psi}_j(P_n) \in \mathcal{Y}, j = 1, \ldots, J) = 1$.*

Let $v \in \{1, \ldots, V\}$ index a sample split into a validation sample $V(v) \subset \{1, \ldots, n\}$ and corresponding training sample $T(v) \subset \{1, \ldots, n\}$ (complement of $V(v)$), where $V(v) \cup T(v) = \{1, \ldots, n\}$, and $\cup_{v=1}^{V} V(v) = \{1, \ldots, n\}$. For each $v \in \{1, \ldots, V\}$, let, $\psi_{njv} \equiv \hat{\Psi}_j(P_{nT(v)})$, $\mathcal{X} \to \mathcal{Y}$, be the realization of the j-th estimator $\hat{\Psi}_j$ when applied to the training sample $T(v)$.

For an observation i let $v(i)$ be the validation sample observation i belongs to, $i = 1, \ldots, n$. Construct a new data set of n observations defined as: (Y_i, Z_i), where $Z_i \equiv (\psi_{njv(i)}(X_i) : j = 1, \ldots, J) \in \mathcal{Y}^J$ is the J-dimensional vector consisting of the J predicted values according to the J estimators trained on the training sample $T(v(i))$, $i = 1, \ldots, n$.

Consider a regression model $z \to m(z \mid \alpha)$ for $E(Y \mid Z)$ indexed by a $\alpha \in \mathcal{A}$ representing a set of functions from $\mathcal{Y}^J$ into $\mathcal{Y}$. Consider a grid (or any finite subset) $\mathcal{A}_n$ of α-values in the parameter space $\mathcal{A}$. Let $K(n) = \mid \mathcal{A}_n \mid$ be the number of grid points which grows at most at a polynomial rate in n: $K(n) \leq n^q$ for some $q < \infty$.

Let

$$\alpha_n \equiv \arg\min_{\alpha \in \mathcal{A}_n} \sum_{i=1}^{n} (Y_i - m(Z_i \mid \alpha))^2.$$

Consider the regression estimator $\psi_n : \mathcal{X} \to \mathcal{Y}$ defined as

$$\psi_n(x) \equiv m((\psi_{jn}(x) : j = 1, \ldots, J) \mid \alpha_n).$$

For each $\alpha \in \mathcal{A}$, define the candidate estimator $\hat{\Psi}_\alpha(P_n) \equiv m((\hat{\Psi}_j(P_n) : j = 1, \ldots, J) \mid \alpha)$: i.e.

$$\hat{\Psi}_\alpha(P_n)(x) = m((\hat{\Psi}_j(P_n)(x) : j = 1, \ldots, J) \mid \alpha).$$

Consider the oracle selector of α*:*

$$\tilde{\alpha}_n \equiv \arg\min_{\alpha \in \mathcal{A}_n} \frac{1}{V} \sum_{v=1}^{V} d(\hat{\Psi}_\alpha(P_{nT(v)}), \psi_0),$$

where

$$d(\psi, \psi_0) = E_0(L(X, \psi) - L(X, \psi_0)) = E_0(\psi(X) - \psi_0(X))^2.$$

For each $\delta > 0$ *we have that there exists a* $C(\delta) < \infty$ *such that*

$$\frac{1}{V} \sum_{v=1}^{V} E d(\hat{\Psi}_{\alpha_n}(P_{nT(v)}), \psi_0) \leq$$

$$(1+\delta) E \min_{\alpha \in \mathcal{A}_n} \frac{1}{V} \sum_{v=1}^{V} d(\hat{\Psi}_\alpha(P_{nT(v)}), \psi_0) + C(\delta) \frac{V \log n}{n}.$$

Thus, if

$$\frac{E \min_{\alpha \in \mathcal{A}_n} \frac{1}{V} \sum_{v=1}^{V} d(\hat{\Psi}_\alpha(P_{nT(v)}), \psi_0)}{\frac{\log n}{n}} \to 0 \text{ as } n \to \infty, \tag{2}$$

then it follows that the estimator $\hat{\Psi}_{\alpha_n}$ *is asymptotically equivalent with the oracle estimator* $\hat{\Psi}_{\tilde{\alpha}_n}$ *when applied to samples of size* $(1 - 1/V)n$*:*

$$\frac{\frac{1}{V} \sum_{v=1}^{V} E d(\hat{\Psi}_{\alpha_n}(P_{nT(v)}), \psi_0)}{E \min_{\alpha \in \mathcal{A}_n} \frac{1}{V} \sum_{v=1}^{V} d(\hat{\Psi}_\alpha(P_{nT(v)}), \psi_0)} \to 1 \text{ as } n \to \infty.$$

If (2) does not hold, then it follows that $\hat{\Psi}_{\alpha_n}$ *achieves the* $(\log n)/n$ *rate:*

$$\frac{1}{V} \sum_{v=1}^{V} E d(\hat{\Psi}_{\alpha_n}(P_{nT(v)}), \psi_0) = O\left(\frac{\log n}{n}\right).$$

Discussion of conditions. The discrete approximation $\mathcal{A}_n$ of $\mathcal{A}$ used in this theorem is typically asymptotically negligible. For example, if $\mathcal{A}$ is a bounded Euclidean set, then the distance between neighboring points on the grid can be chosen as small as $1/n^q$ for some $q < \infty$ so that minimizing a criteria over such a fine grid $\mathcal{A}_n$ versus minimizing over the whole set

$\mathcal{A}$ results in asymptotically equivalent procedures. For example, if α is a Euclidean parameter and $\| m(\cdot \mid \alpha_1) - m(\cdot \mid \alpha_2) \|_\infty < C \| \alpha_1 - \alpha_2 \|$ for some $C < \infty$, where $\| \cdot \|_\infty$ denotes the supremum norm, then it follows that for each $\delta > 0$ we have that there exists a $C(\delta) < \infty$ such that

$$\frac{1}{V} \sum_{v=1}^{V} E d(\hat{\Psi}_{\alpha_n}(P_{nT(v)}), \psi_0) \leq$$

$$(1+\delta) E \min_{\alpha \in \mathcal{A}} \frac{1}{V} \sum_{v=1}^{V} d(\hat{\Psi}_\alpha(P_{nT(v)}), \psi_0) + C(\delta) \frac{\log n}{n},$$

where $\alpha_n = \arg\min_{\alpha \in \mathcal{A}} \sum_{i=1}^{n} (Y_i - m(Z_i \mid \alpha))^2$. The other conclusions of the theorem now also apply.

This theorem implies that the selected prediction algorithm $\hat{\Psi}_{\alpha_n}$ will either perform asymptotically as well (up till the constant) as the best estimator among the family of estimators $\{\hat{\Psi}_\alpha : \alpha \in \mathcal{A}\}$ when applied to samples of size $n(1-1/V)$, or achieve the parametric model rate $1/n$ up till a $\log n$ factor. By a simple argument as presented in van der Laan and Dudoit (2003), Dudoit and van der Laan (2005) and van der Vaart et al. (2006), it follows that by letting the $V = V_n$ in the V-fold cross-validation scheme converge to infinity at a slow enough rate relative to n, then either $\psi_n = \hat{\Psi}_{\alpha_n}(P_n)$ performs asymptotically as well (up till the constant) as the best estimator among the estimators $\{\hat{\Psi}_\alpha : \alpha\}$ applied to the full sample P_n, or it achieves the parametric rate of convergence up till the $\log n$ factor.

The take home message of this theorem is that our super learner will perform asymptotically as well as the best learner among the family of candidate learners $\hat{\Psi}_\alpha$ indexed by α. By choosing the regression model $m(\cdot \mid \alpha)$ so that there exist a α_j so that $m(Z \mid \alpha_j) = Z_j$ for each $j = 1, \ldots, J$ (e.g., $m(Z \mid \alpha) = \alpha Z$), then it follows, in particular, that the resulting prediction algorithm asymptotically outperforms each of the initial candidate estimators $\hat{\Psi}_j$. More importantly and practically, the set of candidate estimators $\hat{\Psi}_\alpha$ can include interesting combinations of these J estimators which exploit the strengths of various of these estimators for the particular data generating distribution P_0 instead of focusing on one of them. For example, if one uses the linear regression model $m(Z \mid \alpha) = \alpha Z$, then the candidate estimators $\{\hat{\Psi}_\alpha : \alpha\}$ include all averages of the J estimators, including convex combinations. As becomes evident in our data analysis and simulation results, the selected super learner ψ_n^* based on a linear (or logistic) regression model is

often indeed (or logistic function of) a weighted average of competing estimators in which various of the candidate learners significantly contribute to the average.

4 Simulation results

In this section, we conducted 3 simulation studies to evaluate the working characteristics of the super learner. These simulations all involve a continuous response variable. For the first simulation, the true model is:

$$\begin{aligned} Y_i = \quad & 2w_1w_{10} + 4w_2w_7 + 3w_4w_5 - 5w_6w_{10} + 3w_8w_9 + w_1w_2w_4 \\ & -2w_7(1-w_6)w_2w_9 - 4(1-w_{10})w_1(1-w_4) + \varepsilon \end{aligned} \tag{3}$$

where $w_j \sim Binomial(p = 0.4)$, $j = 1, \ldots, 10$ and $\varepsilon \sim Normal(0, 1)$. Each observation consists of the 10 dimensional covariate vector W, and the continuous response variable Y. The parameter of interest is $\psi_0(W) = \mathrm{E}_0(\mathrm{Y}|\mathrm{W})$. The simulated learning data set contains a sample of 500 observations (i=1,. . . ,500) from model 3.

We applied the super learner to the learning set using five candidate learners. The first candidate was a simple linear regression model with only main terms, which will be estimated with regular least squares. The second candidate was main terms LARS. Internal cross-validation (i.e. another layer of cross-validation inside each training split) was used the estimate the optimal fraction parameter, $\lambda_0 \in (0, 1)$. The third candidate was the D/S/A algorithm for data-adaptive polynomial regression. For the D/S/A algorithm, we allowed interaction terms and restricted the model to less than 50 terms. The D/S/A uses internal cross-validation to determine the best model in this model space. The fourth candidate was logic regression where the number of trees was selected to be 5 and the number of leaves to be 20 based on 10-fold cross validation of the learning data set. For the logic regression fine-tuning parameters, we searched over #trees $\in \{1, \ldots, 5\}$ and #leaves $\in \{1, \ldots, 20\}$. The final candidate algorithm was random forests. Table 1 contains references for the R packages of each canidate learner.

We applied the super learner with 10-fold cross-validation on the learning set. Applying the prediction to all 10 folds of the learning set gives us the predicted values $Z_i \equiv (\hat{\Psi}_{j\nu(i)}(W_i) : j = 1, \ldots, 5)$ and corresponding Y_i for each observation $i = 1, \ldots, 500$. We then proposed the linear model

method	RMSPE	β_n
Least Squares	1.00	0.038
LARS	1.15	-0.171
D/S/A	0.22	0.535
Logic	0.32	0.274
Random Forest	0.42	0.398
Super Learner	0.20	

Table 2: Simulation Example 1: Estimates of the relative mean squared prediction error (compared to least squares) based on a learning sample of 500 observations and the evaluation sample M=10,000. The estimates for β in the super learner are also reported in the right column ($\alpha_n = -0.018$).

E(Y|Z) $= \alpha + \beta$Z and used least squares to estimate the intercept α and parameter vector β based on (Y_i, Z_i), $i = 1, \ldots, n$.

After having obtained the fit α_n, β_n of α, β, next, each of the candidate learners was fit on the entire learning set to obtain $\hat{\Psi}_j(P_n)(W)$, which gives the super learner $\hat{\Psi}(P_n)(W) = \alpha_n + \beta_n(\hat{\Psi}_j(P_n)(W) : j = 1, \ldots, 5))$ when applied to a new covariate vector W.

To evaluate the super learner next to each of the candidate learners, an additional 10,000 observations are simulated from the same data generating distribution. This new sample is denoted the evaluation sample. Using the models on the learning data set, we calculated the mean squared prediction error (MSPE) on this new evaluation data set for the super learner and each of the candidate learners. Table 2 has the results for the relative mean squared prediction error (RMPSE), where $RMSPE(x) = MSPE(x)/MSPE(least\ squares)$. Among the candidate learners, the D/S/A algorithm appears to have the smallest error, but the super learner improves on the D/S/A fit. The estimates β_n all appear to be nonzero except for the simple linear regression model. The super learner can combine information from the candidate learners to build a better predictor.

The second simulation considers continuous covariates as opposed to binary covariates from the first simulation. Let X be a 20 dimensional multivariate normal random vector and $X \sim N_p(0, 16 * Id_p)$ where $p = 20$ and Id_p is the p-dimensional identity matrix. Each column of X is a covariate in the models used below. The outcome is defined as:

$$Y_i = X_1X_2 + X_{10}^2 - X_3X_{17} - X_{15}X_4 + X_9X_5 + X_{19} - X_{20}^2 + X_9X_8 + \varepsilon, \quad (4)$$

where $\varepsilon \sim Normal(0, 16)$ and X_j is the j^{th} column of X. From this model, 200 observations were simulated for the learning data set and an additional 5,000 were simulated for the evaluation data set similar to the first simulation. The super learner was applied with the following candidate learners:

- Simple linear regression with all 20 main terms.
- LARS with internal cross-validation to find the optimal fraction.
- D/S/A with internal cross-validation to select the best model with fewer than 25 terms allowing for interaction and quadratic terms.
- Ridge regression with internal cross-validation to select the optimal $L2$ penalty parameter.
- Random forests with 1,000 trees.
- Adaptive regression splines.

Table 3 contains the results for the second simulation. As in the first simulation, the relative mean squared prediction error is used to evaluate the candidate learners and the super learner. For this model, simple linear regression, LARS, and ridge regression all appear to have the same results. Random forests and adaptive regression splines are better able to pick up the non-linear relationship, but among the candidate learners, the D/S/A is the best with a relative MSPE of 0.43. But the super learner improves on the fit even more with a relative MSPE of 0.22 by combining the candidate learners. Since the model for $\psi_n^*(z)$ can be near collinear, the estimates of β are often unstable and should not be used to determine the best candidate by comparing the magnitude of the parameter estimate.

The main advantage of the proposed super learner is the adaptivity to different data generating distributions across many studies. The third simulation demonstrates this feature by creating 3 additional studies and applying the super learner and the candidates to all 3 studies then combining the results with the second simulation and evaluating the mean square error across all 4 studies. Equation 5 shows the data generating distributions for the 3 new studies. The data generating distribution for the covariates X is the same as the second simulation example above. To be consistent across the 4 studies, the same candidate learners from the second simulation were applied to these 3 new studies.

method	RMSPE	β_n
Least Squares	1.00	-0.73
LARS	0.91	-0.92
D/S/A	0.43	0.86
Ridge	0.98	0.61
Random Forest	0.71	1.06
MARS	0.61	0.05
Super Learner	0.22	

Table 3: Simulation Example 2: Estimates of the relative mean squared prediction error (compared to Least Squares) based on a learning sample of 200 observations and the evaluation sample M=5,000. The estimates for β in the super learner are also reported in the right column ($\alpha_n = 0.03$).

$$Y_{ij} = \begin{cases} -5 + X_2 + 6(X_{10}+8)_+ - 6(X_{10})_+ - 7(X_{10}-5)_+ \\ \quad - 6(X_{15}+6)_+ + 8(X_{15})_+ + 7(X_{15}-6)_+ + \varepsilon & \text{if } j = 1 \\ 10 \cdot \mathrm{I}(X_1 > -4 \; and \; X_2 > 0 \; and \; X_3 > -4) + \varepsilon & \text{if } j = 2 \\ -4 + X_2 + \sqrt{|X_3|} + sin(X_4) - .3X_6X_{11} + 3X_7 \\ \quad + .3X_8^3 - 2X_9 - 2X_{10} - 2X_{11} + \varepsilon & \text{if } j = 3 \end{cases} \quad (5)$$

where $\varepsilon \sim Normal(0, 16)$ and $\mathrm{I}(x) = 1$ if x is true, and 0 otherwise. For the 4 studies (the 3 new studies combined with the second simulation), the learning sample contained 200 observations and the evaluation sample contained 5,000 observations.

Table 4 contains the results from the second simulation. For the first study ($j = 1$), the adaptive regression spline function is able to estimate well the true distribution. The super learner is not able to improve on the fit, but it does not do worse than the best candidate algorithm. In the second study ($j = 2$), the adaptive regression spline function is not the best candidate learner. The random forests performs best in the second study, but the super learner is able to improve on the fit. The third study ($j = 3$) is similar to the first in that the adaptive regression splines function is able to approximate the true distribution well, but the super learner does not do worse. The squared prediction error from these three studies and the second simulation was combined to give a mean squared prediction error for the four

method	study 1	study 2	study 3	2^{nd} simulation	overall
Least Squares	1.00	1.00	1.00	1.00	1.00
LARS	0.91	0.95	1.00	0.91	0.95
D/S/A	0.22	0.95	1.04	0.43	0.71
Ridge	0.96	0.99	1.02	0.98	1.00
Random Forest	0.39	0.72	1.18	0.71	0.91
MARS	0.02	0.82	0.17	0.61	0.38
Super Learner	0.02	0.67	0.16	0.22	0.19

Table 4: Simulation Example 3: Estimates of the relative mean squared prediction error (compared to least squares) based on the validation sample. The 3 new studies from 5 are combined with the second simulation example and the relative mean squared prediction error is reported in the overall column.

studies. The last column in table 4 gives the relative mspe for each of the candidate learners and the super learner. If the researcher had selected just one of the candidate learners, they might have done well within one or two of the studies, but overall the super learner will outperform the candidate learners. For example, the MARS learner performs well on the first and third study, and does well overall with a relative MSPE of 0.38, but the super learner outperforms the MARS learner with an overall relative MSPE of 0.19. The super learner is able to adapt to the different data generating distributions and will outperform any candidate learner across many studies.

5 Data Analysis

We applied the super learner to the diabetes data set from the LARS package in R. Details on the data set can be found in Efron et al. (2004). The data set consists of 442 observations of 10 covariates (9 quantitative and 1 qualitative) and a continuous outcome. The covariates have been standardized to have mean zero and unit L2 norm. We selected 6 candidate learners for the super learner. The first candidate was least squares using all 10 covariates. Next we considered the least squares model with all possible two-way interactions and quadratic terms on the quantitative covariates. The third and fourth candidates were applying LARS to the main effects and all possible two-way interaction models above. Internal cross-validation was used to select

the "fraction" point for the prediction. The fifth candidate algorithm was D/S/A allowing for two-way interactions and a maximum model size of 64. The final candidate learner was the random forests algorithm. For the super learner, we then used a linear model and estimated the parameters with least squares.

We also applied the proposed super learner to the HIV-1 drug resistance data set in Sinisi et al. (2007) and Rhee et al. (2006). The goal of the data is to predict drug susceptibility based on mutations in the protease and reverse transcriptase enzymes. The HIV-1 sequences were obtained from publicly available isolates in the Stanford HIV Reverse Transcriptase and Protease Sequence Database. Details on the data and previous analysis can be found in Sinisi et al. (2007) and Rhee et al. (2006). The outcome of interest is standardized log fold change in drug susceptibility, defined as the ratio IC_{50} of an isolate to a standard wildtype control isolate; IC_{50} (inhibitory concentration) is the concentration of the drug needed to inhibit viral replication by 50%. We focused our analysis to a single protease inhibitor, nelfinavir, where we have 740 viral isolates in the learning sample of 61 binary predictor covariates and one quantitative outcome.

For the HIV data set, we considered six candidate learners. The first candidate was least squares on all main terms. The second candidate was the LARS algorithm. Internal cross validation was used to determine the best fraction parameter. The third candidate was logic regression. Similar to the simulation example, we used 10-fold cross-validation on the entire learning set to determine the parameters,#trees $\in \{1, \ldots, 5\}$ and #leaves $\in \{1, \ldots, 20\}$, for logic regression. For the HIV data set, we selected #trees $= 5$ and #leaves $= 10$. The fourth candidate was the CART algorithm. We also applied the D/S/A algorithm searching over only main effects terms and a maximum model size of 35. The final candidate was random forests. For the super learner, a linear model was used to estimate the parameters with least squares. All models were fit in R similar to the simulation example above.

To evaluate the performance of the super learner in comparison to each of the candidate learners we split the learning data set into 10 validation data sets and corresponding training data sets. The super learner and each candidate learner was fit one each fold of the cross-validation, giving us a honest cross-validated risk estimate to compare the super learner to each of the candidate learners.

Method	RCV risk	β_n
Least Squares (1)	1.00	0.172
Least Squares (2)	1.13	-0.003
LARS (1)	1.07	0.239
LARS (2)	1.08	0.126
D/S/A	0.98	0.481
Random Forests	1.07	0.027
Super Learner	0.98	

Table 5: Super learner results for the diabetes data set. Least Squares (1) and LARS (1) refer to the main effects only models. Least Squares (2) and LARS (2) refer to the all possible two-way interaction models. Relative 10-fold Honest Cross-Validation risk estimates, compared to main terms least squares (RCV risk) are reported. β_n in the super learner is reported in the last column ($\alpha_n = -6.228$).

5.1 Super Learner Results

Table 5 presents results for the diabetes data analysis. A 10-fold cross-validation estimate of the mean squared error was calculated, and the relative risk estimate is reported. We refer to the cross-validated estimate as honest since we repeating the entire super learner in each fold of the learning data set. The relative cross-validation risk estimate (RCV) is $RCV(x) = CV(x)/CV(main\ terms\ least\ squares)$, where $CV(x)$ is the cross-validation risk estimate for x. Based on the cross validated estimate, the D/S/A has the best estimate among the candidate learners. The super learner does not appear to improve significantly on the D/S/A learner, but it does not do any worse either. We also report the estimates α_n and β_n used in the super learner. The D/S/A algorithm has the largest coefficient (0.481) and appears to be given the most weight in the super learner. We also note that least squares with all possible two-way interactions is barely used in the super learner, with a coefficient of -0.003. This example shows how the super learner can use cross validation to data adaptively select (i.e. give more weight) to the better candidate predictors.

Table 6 presents the results for the HIV data analysis. Based on 10-fold cross validated estimates of the mean squared error, main terms least squares performs best, although random forests and LARS have similar error estimates to least squares. In contrast to the diabetes data analysis above,

Method	RCV risk	β_n
Least Squares	1.00	0.552
LARS	1.03	0.075
Logic	1.52	-0.020
CART	1.77	0.076
D/S/A	1.53	-0.161
Random Forests	1.02	0.510
Super Learner	0.87	

Table 6: Super learner results for the HIV data set. Relative 10-fold honest cross validated risk estimates (RCV risk) compared to least squares are reported. β_n in the super learner is reported in the last column ($\alpha_n = 0.027$).

D/S/A does not perform well on this data set. This highlights the need for a super learner since one candidate algorithm will not work on all data sets. Among the candidate learners, least squares has the smallest cross-validated risk estimate, but the super learner has a smaller risk estimate ($RCV = 0.87$). We also present the estimates for α and β in table 6. Both least squares and random forests appear to be receiving the most weight in the super learner with coefficients 0.552 and 0.510 respectively. Again, the super learner can use the cross validated predictions to data adaptively build the best predictor.

These are both situations where one of the candidate learners does a good job of prediction and gives little room for improvement for the super learner. But these examples also demonstrate that one candidate algorithm may not be flexible enough to perform best on all data generating distributions and since a researcher is unlikely to know *a priori* which candidate learner will work best, the super learner is a natural choice for prediction.

6 Discussion.

The new super learning approach provides both a fundamental theoretical as well as practical improvement to the construction of a predictor. The super learner is a flexible prediction algorithm which can perform well on many different data generating distributions, and utilizes cross-validation to protect against over-fitting. We wish to stress that the theory suggests that to achieve the best performance one should not apply this algorithm

to a restricted set of candidate learners, but one should aim to include any available sensible learners. In addition, the amount of computations does not exceed the amount of computations it takes to calculate each of the candidate learners on the training and full data sets. In our simulations we used a particular set of available learners only because they were easily available as R functions. Thus, the potential for improving learners applies to a very wide array of practical problems.

Our results generalize to parameters which can be defined as minimizers of a loss function, including (unknown) loss functions indexed by parameters of the true data generating distribution (van der Laan and Dudoit (2003)). In particular, the super learner approach applies to maximum likelihood estimation in semiparametric or nonparametric models for the data generating distribution, and to targeted maximum likelihood estimation with respect to a particular smooth functional of the density of the data, as presented in van der Laan and Rubin (2007).

7 Appendix

Under the Assumption A1 that the loss function $L(O,\psi) = (Y - \psi(X))^2$ is uniformly bounded, and the Assumption A2 that the variance of the ψ_0-centered loss function $L(O,\psi) - L(O,\psi_0)$ can be bounded by its expectation uniformly in ψ, van der Laan et al. (2006) (Theorem 3.1) establish the following finite sample inequality.

Theorem 2. *Let* $\{\hat{\psi}_k = \hat{\Psi}_k(P_n), k = 1, ..., K(n)\}$ *be a given set of* $K(n)$ *estimators of the parameter value* $\psi_0 = \arg\min_{\psi\in\Psi} \int L(o,\psi) dP_0(o)$. *Let* $d_0(\psi,\psi_0) \equiv E_{P_0}\{L(O,\psi) - L(O,\psi_0)\}$ *denote the risk difference between a candidate estimator* ψ *and the parameter* ψ_0. *Suppose that* $\mathbf{\Psi}$ *is a parameter space so that* $\hat{\Psi}_k(P_n) \in \mathbf{\Psi}$ *for all* k, *with probability 1. Let* $\hat{K}(P_n) \equiv \arg\min_k E_{B_n} \int L(o, \hat{\Psi}_k(P^0_{n,B_n})) dP^1_{n,B_n}(o)$ *be the* cross-validation selector, *and let* $\tilde{K}(P_n) \equiv \arg\min_k E_{B_n} \int L(o, \hat{\Psi}_k(P^0_{n,B_n})) dP_0(o)$ *be the comparable* oracle selector. *Let* p *be the proportion of observations in the validation sample. Then, under assumptions A1 and A2, one has the following finite sample inequality for any* $\lambda > 0$ *(where* $C(\lambda)$ *is a constant, defined in van der Laan*

et al. (2006)):

$$Ed_0(\hat{\Psi}_{\hat{K}(P_n)}(P^0_{n,B_n}), \psi_0) \leq$$
$$(1+2\lambda)Ed_0(\hat{\Psi}_{\tilde{K}(P_n)}(P^0_{n,B_n}), \psi_0) + 2C(\lambda)\frac{1+log(K(n))}{np}.$$

References

L. Breiman. Random Forests. *Machine Learning*, 45:5–32, 2001.

L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. *Classification and Regression Trees*. The Wadsworth Statistics/Probability series. Wadsworth International Group, 1984.

S. Dudoit and M. J. van der Laan. Asymptotics of cross-validated risk estimation in estimator selection and performance assessment. *Statistical Methodology*, 2:131–154, 2005.

B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. Least Angle Regression. *Annals of Statistics*, 32(2):407–499, 2004.

Y. Freund, R. E. Schapire, Y. Singer, and M. K. Warmuth. Using and combining predictors that specialize. In *Twenty-Ninth Annual ACM Symposium on the Theory of Computing*, 1997.

J. H. Friedman. Multivariate adaptive regression splines. *Annals of Statistics*, 19(1):1–141, 1991.

J. V. Hansen. Combining predictors: Some old methods and a new method. In *ICCIN*, 1998. URL `citeseer.ist.psu.edu/article/hansen98combining.html`.

A.E. Hoerl and R.W. Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(3):55–67, 1970.

S. Rhee, J. Taylor, G. Wadhera, J. Ravela, A. Ben-Hur, D. Brutlag, and R. W. Shafer. Genotypic predictors of human immunodeficiency virus type 1 drug resistance. *Proceedings of the National Academy of Sciences USA*, 2006.

I. Ruczinski, C. Kooperberg, and M. LeBlanc. Logic Regression. *Journal of Computational and Graphical Statistics*, 12(3):475–511, 2003.

S. E. Sinisi and M. J. van der Laan. Deletion/Substitution/Addition algorithm in learning with applications in genomics. *Statistical Applications in Genetics and Molecular Biology*, 3(1), 2004. Article 18.

S. E. Sinisi, E. C. Polley, S.Y. Rhee, and M. J. van der Laan. Super learning: An application to the prediction of HIV-1 drug resistance. *Statistical Applications in Genetics and Molecular Biology*, 6(1), 2007.

M. J. van der Laan and S. Dudoit. Unified Cross-Validation Methodology for Selection Among Estimators and a General Cross-Validated Adaptive Epsilon-Net Estimator: Finite Sample Oracle Inequalities and Examples. Technical Report 130, Division of Biostatistics, University of California, Berkeley, Nov. 2003. URL `http://www.bepress.com/ucbbiostat/paper130/`.

M. J. van der Laan and D. Rubin. Targeted maximum likelihood learning. *International Journal of Biostatistics*, 2(1), 2007.

M. J. van der Laan, S. Dudoit, and A. W. van der Vaart. The cross-validated adaptive epsilon-net estimator. *Statistics and Decisions*, 24(3):373–395, 2006.

A.W. van der Vaart, S. Dudoit, and M.J. van der Laan. Oracle inequalities for multi-fold cross vaidation. *Statistics and Decisions*, 24(3), 2006.